Describing the Design Contributors to Mode Error

Asaf Degani[1] and Alex Kirlik[2]
[1]NASA Ames Research Center, Moffett Field, CA USA
adegani@mail.arc.nasa.gov

[2]Center for Human-Machine Systems Research
School of Industrial & Systems Engineering
Georgia Institute of Technology, Atlanta, GA USA
kirlik@chmsr.gatech.edu

Abstract

Although mode error has attracted a great deal of recent interest from those involved with complex systems, the design factors that contribute to this problems are not well understood. In this paper we provide an introduction to a modeling framework we use to describe complex, multi-modal human-machine systems. The modeling approach is based on the statechart formalism, which is an extension of the finite state machine formalism to allow representation of concurrency, hierarchy, default transitions, and broadcast of parameter information. After having used this approach to model a number of human-machine systems known to create mode control problems for the operator, we have identified a number of system design features that contribute to mode error.

Introduction

The purpose of this paper is to describe a methodololgy we have developed for the analysis and design of the interface to complex, high-technology control systems. This methodology has been motivated by our experiences over the past 10 years conducting cognitive engineering research in domains such as commercial aviation, military command and control, and emergency medical systems in the health care industry. In domains such as these, responsibility for system control is shared by human operators and a variety of semi-automated control systems (e.g., autopilots, decision support systems, automatic blood pressure measurement devices). We have observed that a key contributor to operator confusion and error in these systems is the design of the interface between the human operator and the semi-automated control systems which jointly share responsibility for effective system performance.

In particular, these interfaces often provide insufficient information to allow the operator to correctly determine the status of the automated control systems. The methodology we will describe involves creating a graphical model of the entire human-task-display-control-plant system, and subsequently examining this model to identify certain patterns that are known to contribute to operator confusion and error. We suggest that this methodology can be used to evaluate a candidate interface design early in the design cycle so that any design deficiencies can be remedied before the system is fielded and operated. Prospects for partially automating this analysis methodology using computer software will also be discussed.

Mode Confusion and Mode Error

Many complex semi-automated control systems contain modes. A mode can be considered to be a state of a device or a state of a control system which determines its manner of behaving. Everyday examples include degree/radian mode on a hand calculator, and TV/Video mode on a television. In complex control systems, a mode may determine the dynamic behavior of a system over an extended time period (e.g., cruise control in an automobile, autoflight modes on a modern jetliner). Modes and the interaction problems caused by mode-based systems have long been of interest to designers of computer interfaces and everyday devices (see Norman, 1988).

Mode confusion exists when an operator does not know the mode state of the system with which he or she is interacting. Mode error results when innappropriate actions are taken due to this confusion. Mode errors have been cited as the cause of a number of incidents and accidents in high-technology systems (see Degani, 1996, for a discussion). The potential for mode error is created whenever designers introduce control automation that has a variety of possible states or manners of dynamically behaving. The central human-machine interface (HMI) design problem

0-8186-8341-4/98 $10.00 © 1998 IEEE

created by such systems is to ensure that the interface contains the information necessary for the operator to correctly identify the underlying state of the automated systems with which the human operator cooperates to control the target system.

Describing the Design Contributors to Mode Error

Effectively designing the iinterface to mode-based systems presents considerable subtleties due to the often complex interactions between interface displays and controls, semi-automatic control mechanisms, and the plant being controlled. Our goal in developing a technique for the analysis of such systems was to put forth a modeling approach with resources to allow the analyst to graphically document these complex interactions. This technique then allows the analyst to subsequently examine the resulting graphical model to identify particular patterns of interactions known to cause mode confusion and mode error. To do so, we adopted the *statechart* modeling formalism developed by Harel (1987). Statecharts can be viewed as an extension of finite state machine (FSM) transition diagrams to allow description of hierarchy, default transitions, concurrency in the behavior of multiple subsystems, and the broadcast of information among subsystems. Due to these embellishments, statecharts often allow a visually simpler representation of the complex interactions among various subsystems in a multi-component dynamic system. Full details of the modeling approach are presented in the recent disseration by Degani (1996).

Figure 1 depicts the design of a simple home thermostat in both FSM and statechart modeling formalisms. Note that the statechart representation has fewer transition links than does the FSM representation. In this case, this reduction in links is due to the use of hierarchy (the inner box) to represent a subsystem of the thermostat, and the use of a default transition (in the upper left corner of the box) to represent the initial state occupied by the subsystem when it is engaged, in this case by the activation of an

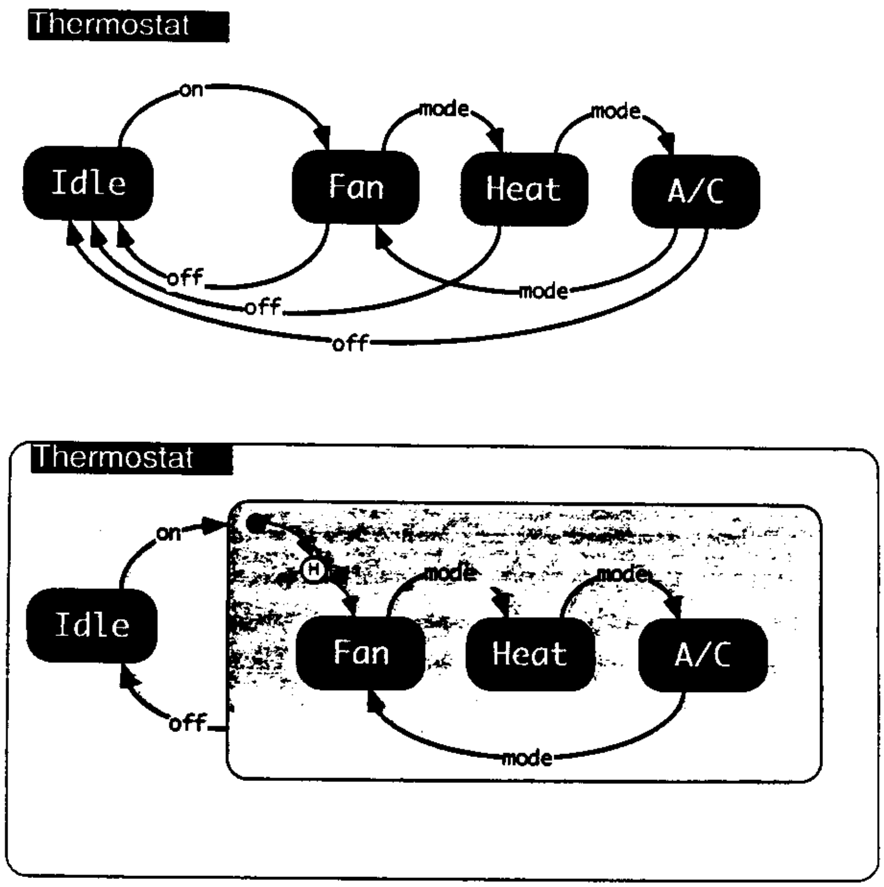

Figure 1. A Home Thermostat in both FSM and Statechart Representations

"on" switch. The reduction in visual complexity enabled by statecharts is relatively modest in this example. However, the use of techniques to represent hierarchy, defaults, concurrency and broadcast in statecharts are crucial in modeling more complex systems. These techniques provide the resources necessary to defeat the combinatorial increase in the number of state transitions caused by increased numbers of interacting subsystems, as are often found in human-machine systems containing semi-automated control systems.

For example, consider Figure 2, which is a statechart representation of a standard cruise control system in a modern automobile. This model clearly shows five different levels of human versus machine involvement in the control of automobile speed, progressing from level 1, in which the human is in full control, to level 5, in which the machine is in full control. To clearly appreciate how statecharts have allowed for the reduction of visual complexity in this case, the reader is invited to represent the same transition information presented in Figure 2 in terms of a non-hierarchical or "flat" FSM representation. Note also how this model highlights a deficiency in the HMI design for this system. The transition represented by the broken arrow between levels 4 and

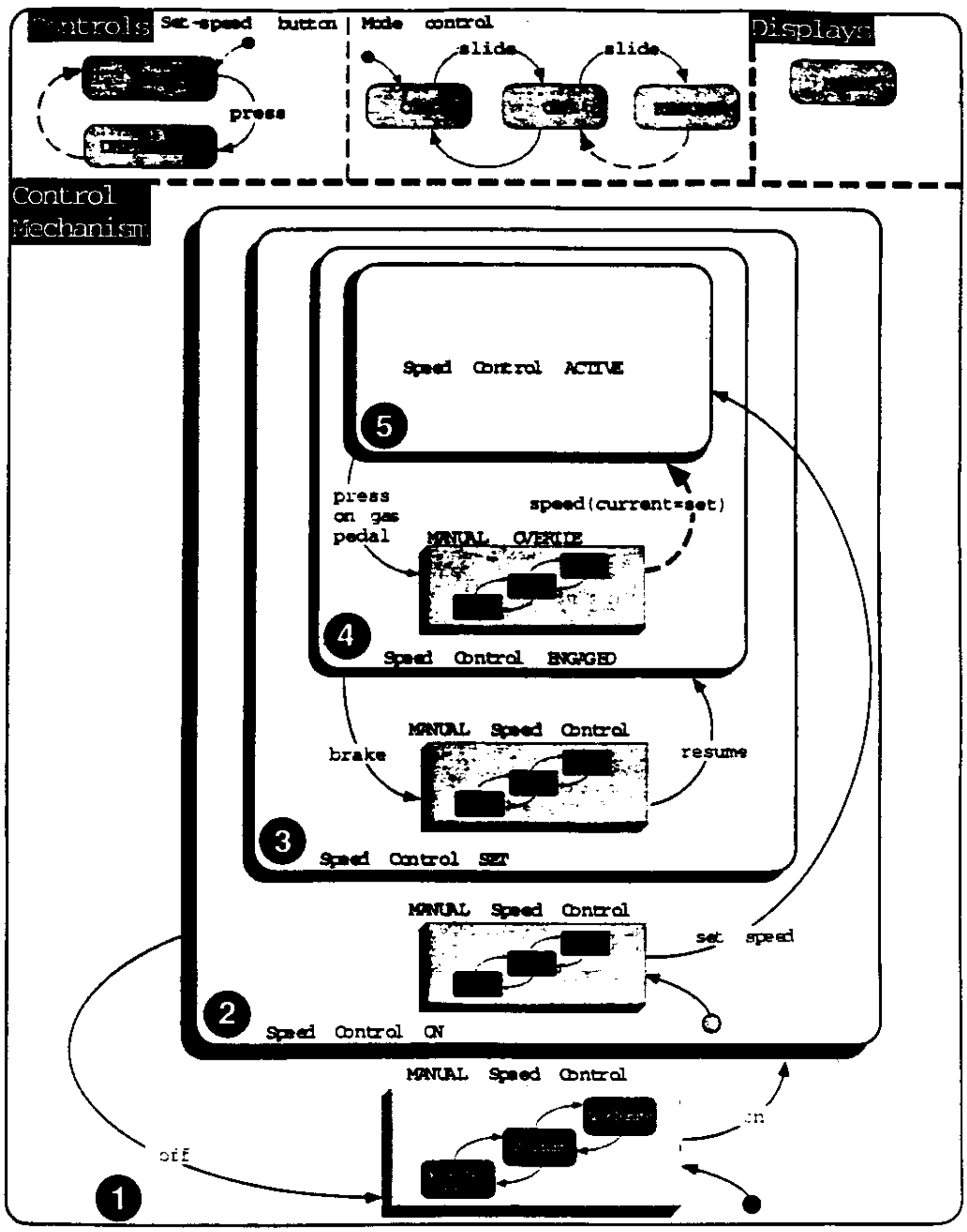

Figure 2. A Statechart Model of Cruise Control

5 is an automatic transition from manual override of the cruise control (caused by the operator depressing the gas pedal) and re-engagement of the cruise control (caused by the vehicle subsequently slowing to the commanded speed), which has been "remembered" by this semi-automated system. Before this automatic transition, the system is in a state which is perceptually indistinguishable from level 3, in which the operator's braking action has completely disengaged the cruise control (rather than merely making it "dormant" as in the case of pressing the gas pedal to override the system). A lack of HMI information capable of differentiating between these disengaged and dormant states has been known to cause mode confusion and mode errors for drivers. In such cases, the automobile unexpectedly (and dangerously) accelerates after re-slowing to the commanded speed at some point after the driver has depressed the gas pedal to override the cruise control system.

Conclusion

We have constructed and examined a variety of statechart models of semi-automated human-machine systems where mode confusion and mode error are known to have occurred. In all cases we have been able to identify features of the statechart models associated with HMI design problems contributing to these confusions and errors (see Degani, 1996, for a complete presentation). These features are: 1. Underspecification of system state caused by multiple plant states with only a single display state; 2. The presence of an automatic transistion within the semi-automated control system which is not reflected in the HMI; 3. The presence of a default transition within the semi-automated control system which is not reflected in the HMI; 4. The presence of operator-induced transitions which are not reflected in the HMI; 5. The presence of circular mode transitions not reflected in the HMI, and 6. Transitions among modes of semi-automated control systems caused by programmed reference values (or control parameters) which are not reflected in the HMI.

Statecharts provide a modeling formalism which requires an analyst to describe a human-machine system, including HMI displays and controls, in great detail. After the model is constructed, the analyst can then assess the adequacy of HMI design by looking for any of the conditions known to contribute to mode confusion and error. HMI design deficiencies can then be remedied before the system is constructed, fielded, and operated. We are currently considering how one could use available statechart software modeling tools, currently used by control engineers for specifying the fully automatic components of control systems, to also assist in the analysis and evaluation of the control functions performed by human operators in concert with control automation.

References

Degani, A. (1996). *Modeling Human-Machine Interaction: Modes, Models, and Patterns of Interaction.* Unpublished Ph.D. dissertation, Center for Human-Machine Systems Research, School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8, 231-274.

Norman, D.A. (1988). *The psychology of everyday things.* New York: Basic Books.